COMPSCI 2C03 : Assignment 1

Fall 2022

Nicholas Moore

Maximum Grade: 43/36

Released: September 21, 2022 Due Date: October 5, 2022 at 11:59PM

Instructions

- Submissions will be made through the Avenue assignment 1 dropbox folder.
- This assignment contains an optional coding exercise, which will require the submission of program source code. Please zip together all files you wish to submit, name the zip file "A1_¡MACID¿.zip", where ¡MACID¿ is your MacID. Submit this zip file to Avenue. Not doing this and just submitting a bunch of files is the equivalent of not stapling your assignment before handing it to the teacher.
- When submitting programming code, please test it before submission. Code must run to be correct.
- Late submissions will not be accepted except in the gravest of circumstances.
- One bonus mark is available for submission of the non-coding components of this assignment as a LaTeXgenerated pdf. To be eligible you must also submit your LaTeXsource file.
- Don't cheat or I'll catch you.

Questions

Q1 : Parenthesis Matching (12 + 3 points)

Your task is to write and analyse an algorithm to verify parenthesis matching in a mathematical expression. Your algorithm must take as input a mathematical expression expressed as a string, and output a boolean result. True indicates that there are no mismatched parentheses in the input string. False indicates that a parenthesis mismatch occurs in the string. The recommended data structure for this problem is a **stack**.

1. (8 points) Write your algorithm in pseudocode.
2. (4 points) Analyse your algorithm and give its worst-case runtime complexity.
3. (3 bonus points!) Encode this algorithm in Java. In this and all future questions, the underlying data structure (in this case a stack) must have its own class. You are free to implement the stack however you like, so long as the API is preserved.

Q2 : Big O(meg(thet)a) (12 points)

For each of the following:

$$\frac{65n^4 + 2n + 3}{n + 1} \in \Theta(n^3) \tag{1}$$

$$45n\text{log}n + 2n + 1 \in \Theta(n\text{log}n) \tag{2}$$

$$n^2 \notin \Theta(\text{log}n) \tag{3}$$

$$n^n \notin \Theta(2^n) \tag{4}$$

1. (2 points each) Demonstrate the statement using only the definitions of $O(f(n))$, $\Omega(f(n))$, and $\Theta(f(n))$.
2. (1 point each) Demonstrate the statement using limit rules.

Selection Sort (12 + 3 points)

Adapt the Selection Sort algorithm given in class (Topic 4) to operate over a doubly linked list. For the purposes of pseudocode, you may assume access to the full DLL API, and may assume the class maintains pointers to the head and tail of the list.

- (8 points) Write this algorithm as pseudocode.
- (4 points) Give a worst-case runtime complexity for your algorithm.
- (3 bonus points!) Encode this algorithm in Java. You may also need to implement a doubly linked list class if you don't already have one kicking around.